

btrfs and snapper

The Next Steps From Pure filesystem
Features to Management Integration and Compliance

SUSECon 2012, Orlando, Florida

Gábor Nyers

System Engineer @SUSE
gnyers@suse.com

Arvin Schnell

Senior Software Developer
aschnell@suse.com



Abstract

"btrfs" as a filesystem has been getting a lot of attention over the past few years. While it is interesting from its feature set alone (checksums, copy on write, snapshots, Volume Management integration), some of these features are not directly useful for customers without a proper integration into userland tools, management infrastructure and compliance processes.

This session will help you learn how to use snapper for snapshot management on SUSE Linux Enterprise. We will provide an outlook for future functionality of snapper and integration of btrfs into the system.

Agenda

Introduction to Btrfs

Btrfs in SUSE distro's
Snapper

Btrfs in-depth
Use cases

Summary and
Questions

- Btrfs specs
- Features and Concepts
- Current limitations
- Support from distributions

What People Say About Btrfs...

Chris Mason (lead developer Btrfs)

- General purpose filesystem that scales to very large storage
- Focused on features that no other Linux filesystems have
- Easy administration and fault tolerant operation

Ted Tso (lead developer Ext4)

- (Btrfs is) "... the way forward"

Others:

- "Next generation Linux filesystem"
- "Btrfs is the Linux answer to ZFS"

Why Another Linux filesystem?

- Solve Storage Challenges
 - Scalability
 - Data Integrity
 - Dynamic Resources (expand and shrink)
 - Storage Management
 - Server, Cloud – Desktop, Mobile
- Compete with and exceed the filesystem capabilities of other Operating Systems

Btrfs Specs

- Max volume size : 16 EB (2^{64} byte)
- Max file size : 16 EB
- Max file name size : 255 bytes
- Characters in file name : any, except 0x00
- Directory lookup algorithm : B-Tree
- Filesystem check : on- and off-line
- Compatibility
 - POSIX file owner/permission
Access Control Lists (ACLs)
Asynchronous and Direct I/O
 - Hard- and symbolic links,
Extended Attributes (xattrs),
Sparse files

A Few Btrfs Concepts

- B-Tree
 - Index data structure
 - Fast search, insert, delete
- Subvolume
 - Filesystem inside the filesystem
 - Independent B-Tree linked to some directory of the root subvolume
- Metadata
 - “normal” metadata: size, Inode, atime, mtime, etc...
 - B-Tree structures
- Raw data
 - Actual content of files

Btrfs Feature Summary 1/2

- **Extents**
 - Use only what's needed
 - Contiguous runs of disk blocks
- **Copy-on-write**
 - Never overwrite data!
 - Similar to CoW in VMM
- **Snapshots**
 - Light weight
 - At file system level
 - RO / RW
- **Multi-device Management (no L3 yet)**
 - mixed size and speed
 - on-line add and remove devs
- **Object level RAID:**
 - 0, 1, 10
- **Efficient small file storage**
- **SSD support (optimizations, trim)**

Btrfs Feature Summary 2/2

- Checksums on data and meta data
- On-line:
 - Balancing
 - Grow and shrink(!)
 - Scrub
 - Defragmentation
- Transparent compression (gzip, lzo)
- In-place conversion from Ext[34] to Btrfs
- **Send/Receive**
 - Similar to ZFS' send/receive function
- **Seed devices**
 - Overlay a RW file system on top of an RO

Btrfs Planned Features

- Quota support
 - Aug 2012: 1st implementation available
- Object-level RAID 5, 6
- Data de-duplication:
 - On-line de-dup during writes
 - Background de-dup process
- Tiered storage
 - Frequently used data on SSD(s)
 - “Archive” on HDD(s)



Btrfs file system check, recovery and repair

- Status of btrfsck
 - Released in SLES 11 SP2 and OL6 with UEK2
 - Off-line filesystem repair
- Auto recovery at mount
 - mount -o recovery
- Btrfs-restore
 - Read-only recovery tool

Btrfs Features:

Current limitations (Aug 2012)

- Full featured off-line fsck repair tool, however:
 - First implementation of off-line fsck already available
 - On-line repair options with btrfs scrub
 - Recovery mount option
 - btrfs-restore utility
- Limited bootloader support (GRUB2 only)
- Quota support (unreleased)
 - Tools to help set up and report subvolume sizes
- Limited nr. of hard links to a file 3 – 250 (patch)
- RAID 5 and 6 (patch)
- Consistent documentation
 - Btrfs Wiki @ kernel.org is available again

Distro Support Status

Support

- SUSE® Linux Enterprise Server **11 SP2**
- OpenSUSE 11.4, 12.1
- Oracle Linux 6 with **UEK2**
- Debian 6
- Ubuntu **11.04**

Technology Preview / unsupported

- Red Hat Enterprise Linux
- Fedora
- and others...

Agenda

Introduction to Btrfs

Btrfs in SUSE distro's Snapper

Btrfs in-depth
Use cases

Summary and
Questions

- Btrfs integration in SLES and openSUSE
- Partitioner
- Planned features
- Snapshot management with Snapper

Btrfs integration in SLE 11 SP2 and openSUSE 12.1

Basic integration into

- Installer
 - Btrfs as root file system
 - Recommendation for subvolume layout
- Partitioner
 - Create Btrfs
 - Create subvolumes

Tools

- Snapper
 - Manage snapshots
 - Automatically create snapshots
 - Display differences between snapshots
 - Roll-back

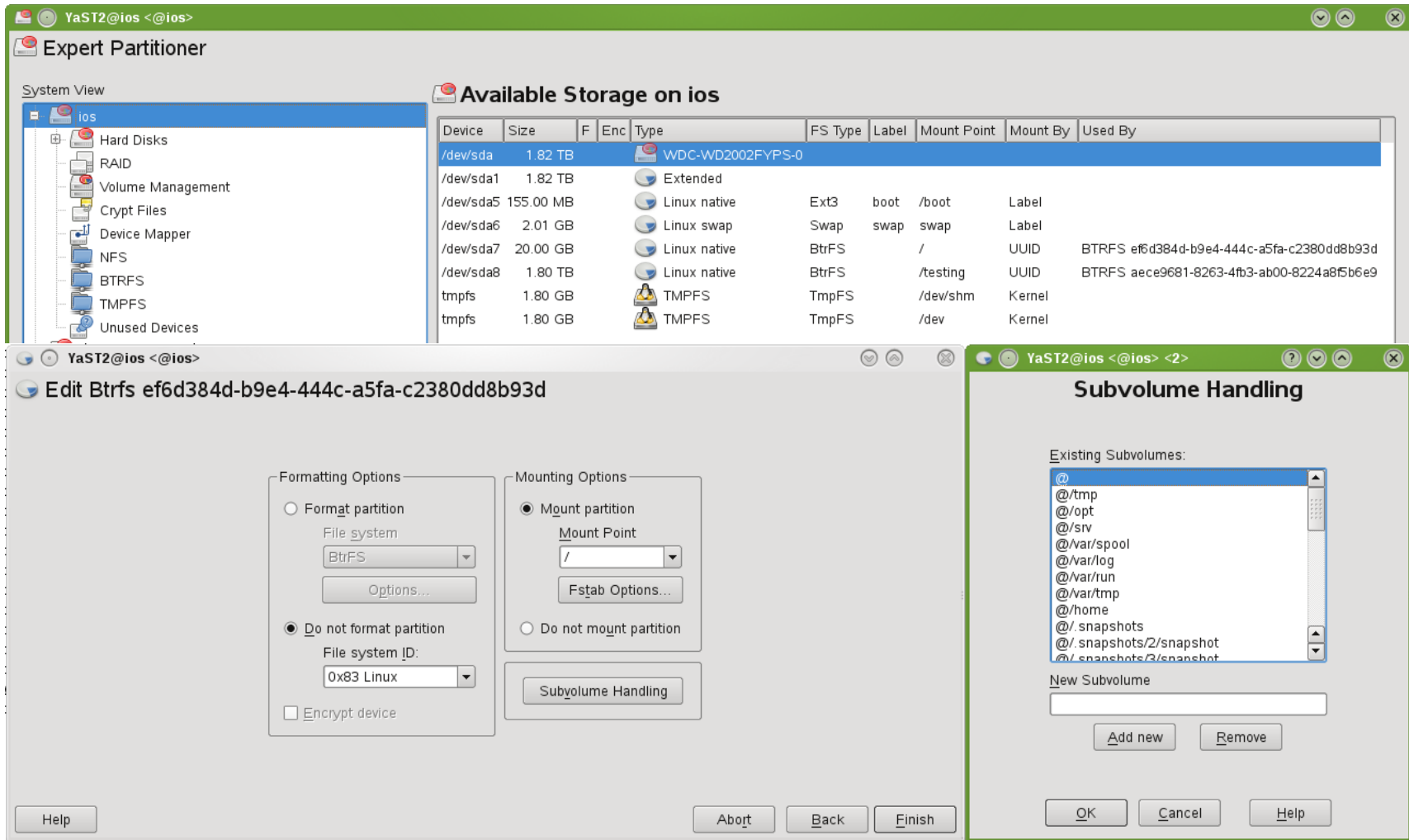
Btrfs integration in SLE 11 and openSUSE

Future plans

- YaST partitioner support for:
 - Built-in multi-volume handling and RAID
 - Transparent compression
- Btrfs support in AutoYaST
- Bootloader support for /boot on btrfs
- Snapshot creation as non-root user (DBus support)



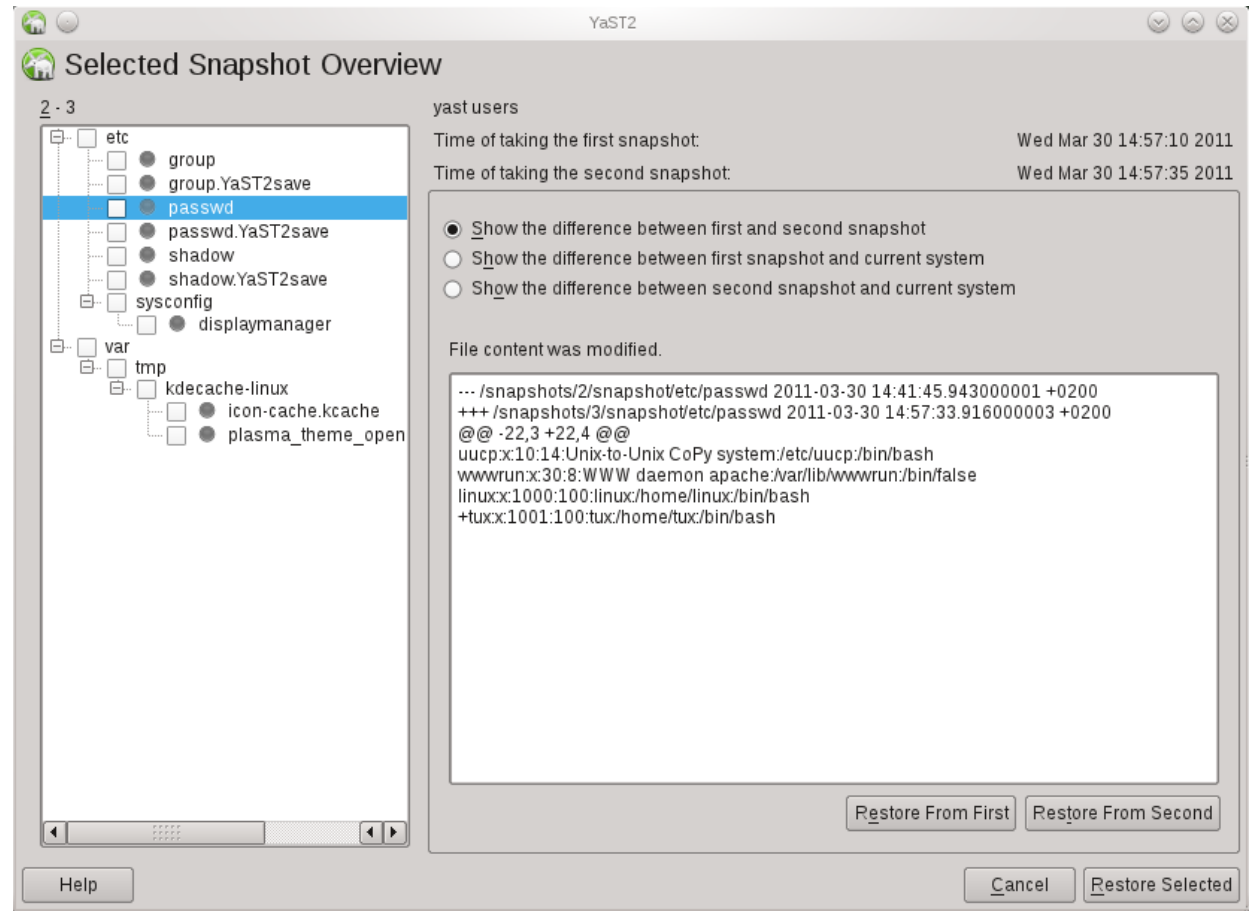
Btrfs integration in YaST Partitioner



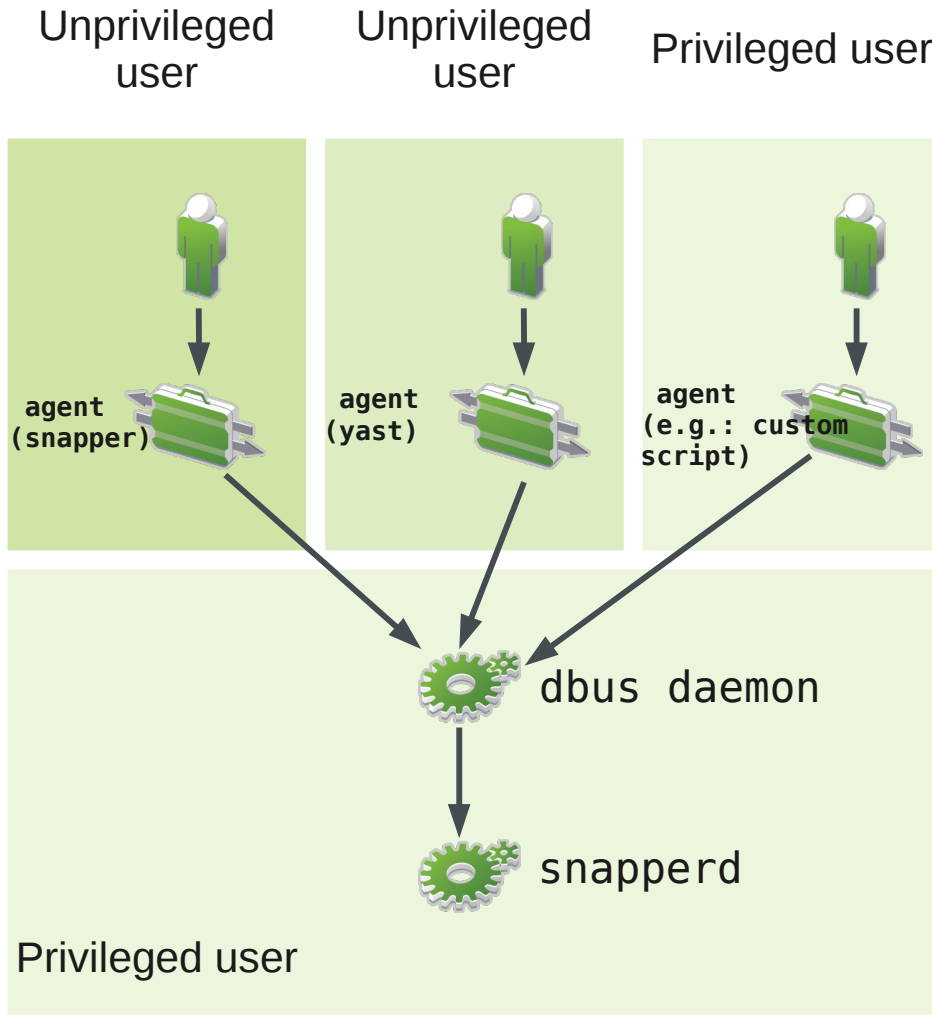
Snapshot management with Snapper

Functions

- Automatic snapshots
- Integration with YaST and Zypp
- Rollback
- Integration points



Snapper future: DBus support



- Snapper is split up:
 - snapper (client)
 - snapperd (server)
- Authorized users submit request through DBus
- snapperd performs actions on behalf of users
- Authorization scheme
 - Users
 - Agents

Demo 1

Snapper

- Snapper module for YaST
- Snapper integration with YaST
- Snapper command line tool
- Snapper as non-root

Snapper - Metadata

Meta information stored with each snapshot:

- **Type** : [Pre | Post | Single]
- **#** : Nr of snapshot
- **Pre #** : Matching “Pre” number, if type is “Post”
- **Date** : Timestamp
- **User** : User who created the snapshot
- **Cleanup** : Cleanup algorithm for this snapshot
- **Description** : A fitting description of the snapshot (free text)
- **Userdata** : key=value pairs to record all sorts of useful information about the snapshot in an (e.g.: easily parsing from scripts)

Agenda

Introduction to Btrfs

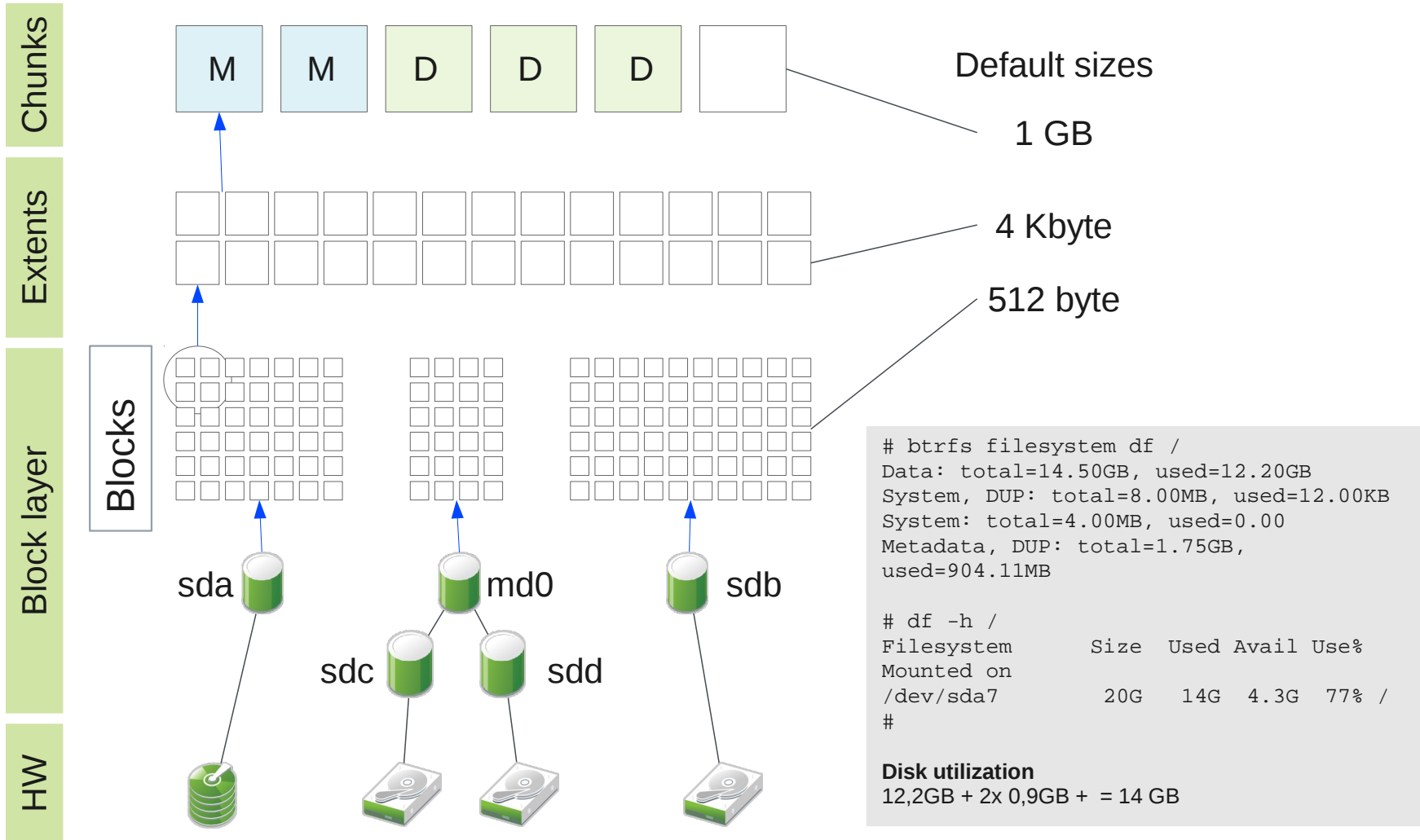
Btrfs in SUSE distro's Snapper

Btrfs in-depth
Use cases

Summary and
Questions

- In-depth
 - Extents, Copy-on-Write, Subvolumes, Snapshots
- Recommendations
- Use cases
- Performance
- Demo

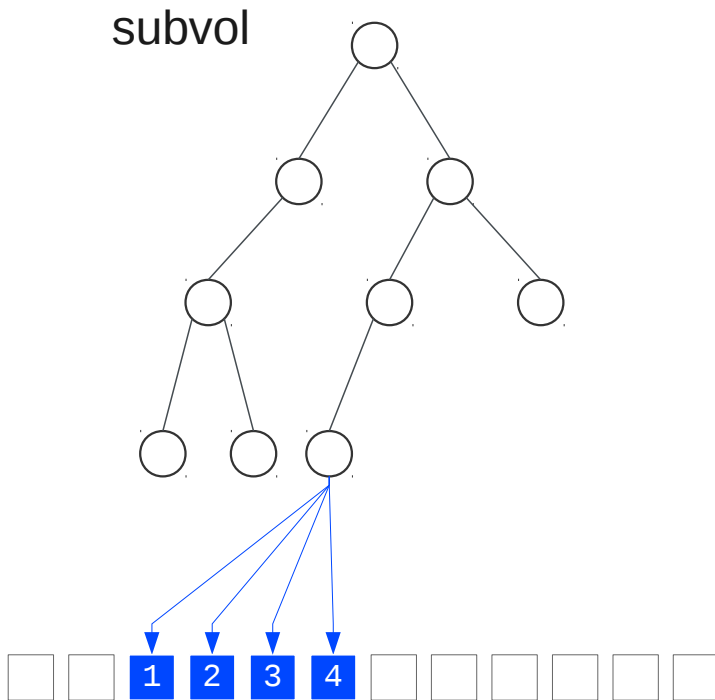
In-depth Btrfs: Extents and Storage Organization



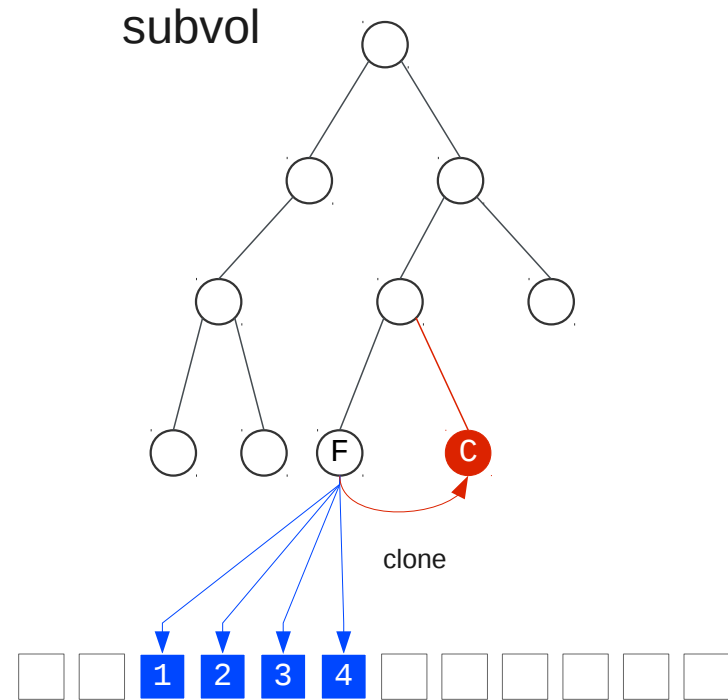
Btrfs Features:

Copy on Write explained 1/4

1



2

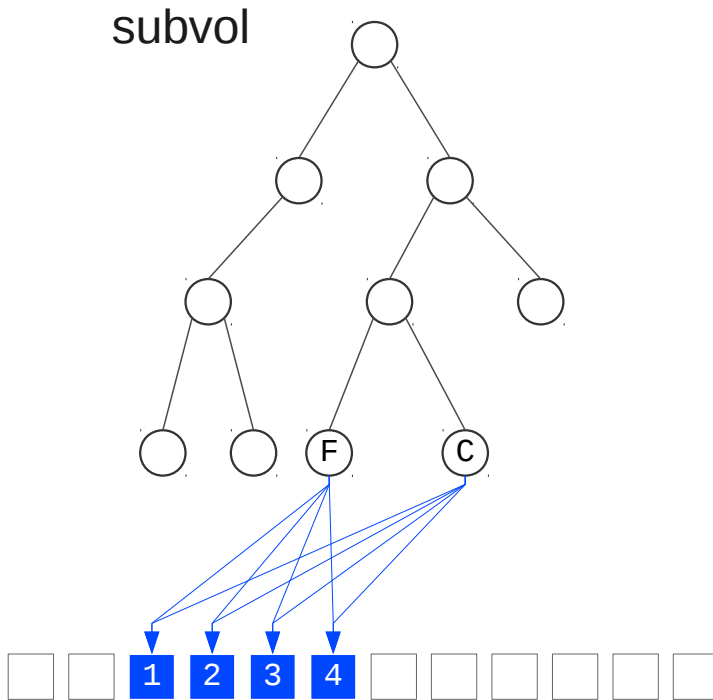


Copy Clone
cp --reflink=always F C

Btrfs Features:

Copy on Write explained 2/4

3

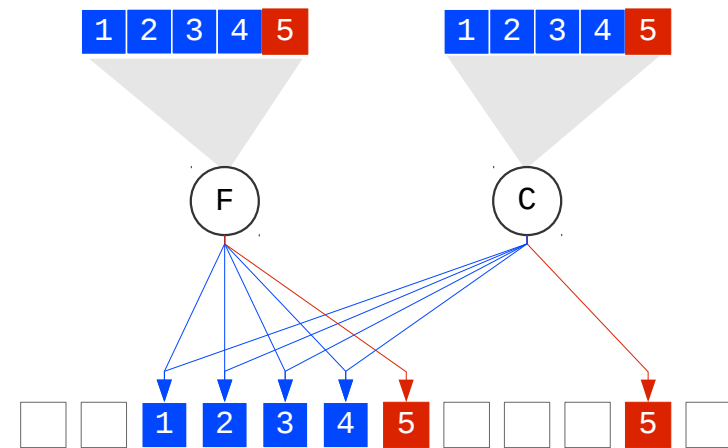


4

Append to files:

F: +1 extent

C: +1 extent



Btrfs Features:

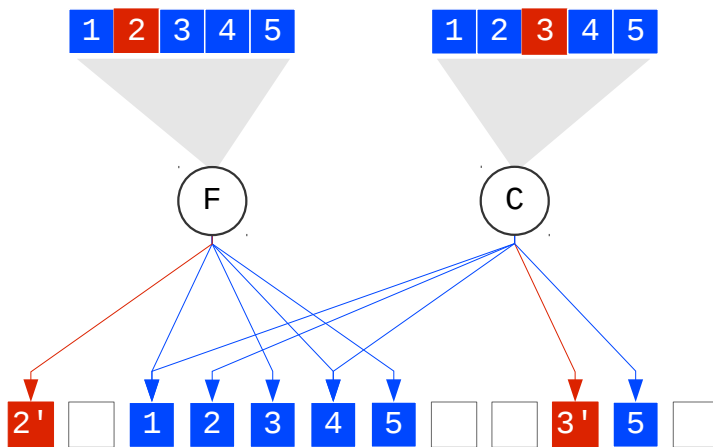
Copy on Write explained 3/4

5

Modify extent:

F: 2

C: 3

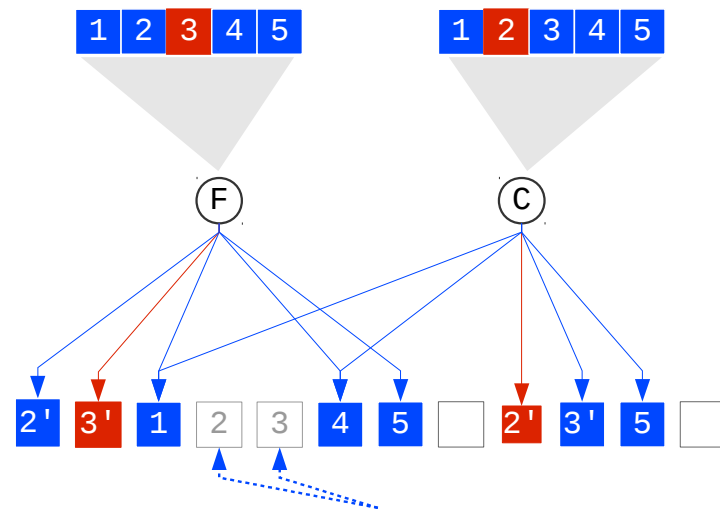


6

Modify extent:

F: 3

C: 2



extents needing “trimming”
('discard' mount option)

Btrfs Features:

Copy on Write explained 4/4

7

Truncate files:

F: -2 extent

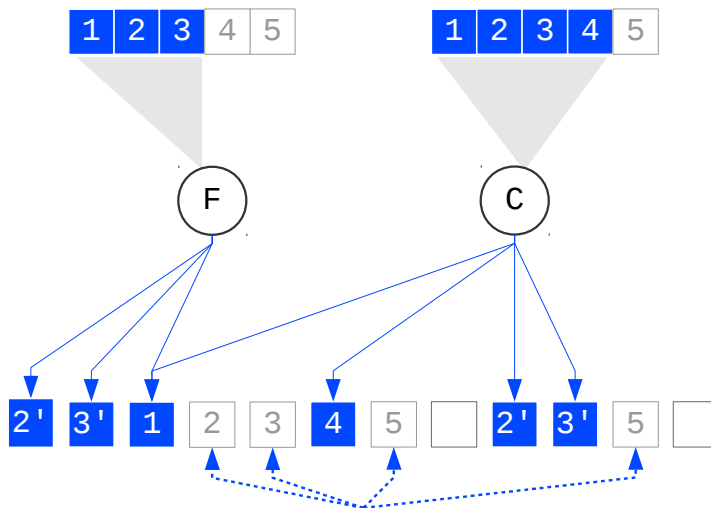
C: -1 extent

8

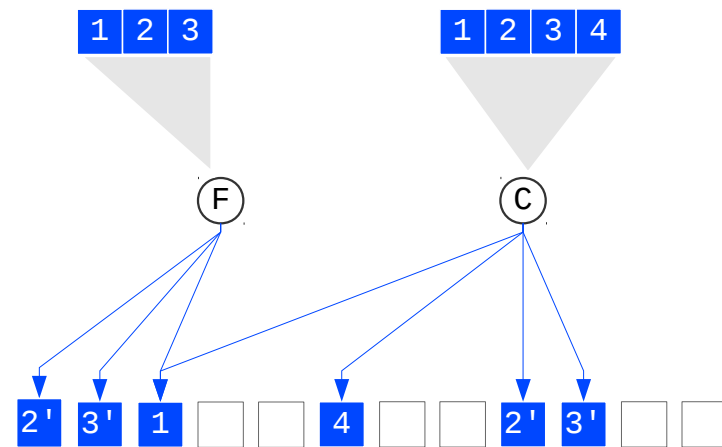
Trim command:

ATA : `man 8 fstrim`

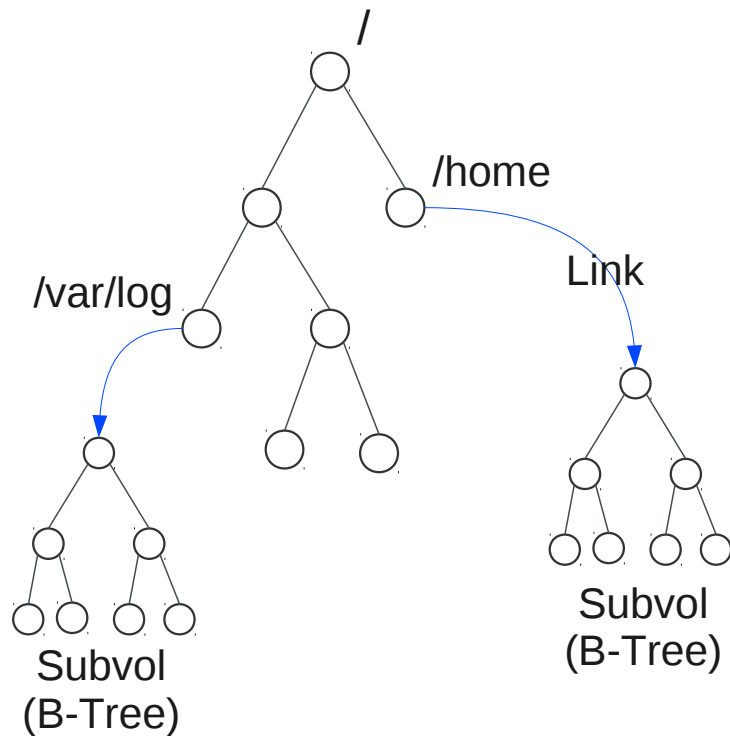
SCSI: `man 8 sg_unmap`



extents needing "trimming"
(`'discard'` mount option)

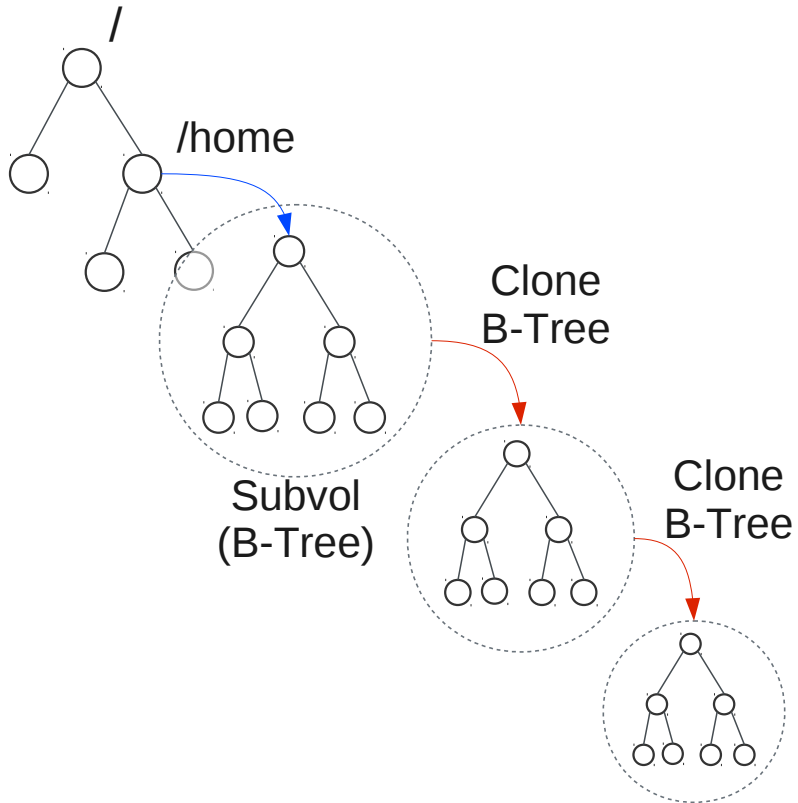


In-depth Btrfs: Subvolumes



- Independent B-Tree linked to some directory of the root subvolume
- A part of the file system
- Appears on file system as a directory
- Subvols on a Btrfs file system share the same device pool
- Independently snapshotable (copy B-Tree)
- Independently mountable

In-depth Btrfs: Snapshots



- A record of the state of a subvolume
- CoW copy of another subvolume
- After creation, snapshot shares all raw data and metadata with parent
- (practically) unlimited in number
- Read Only, Writable and Nested (= “snapshot of a snapshot”)
- Snapshots on the file system level

In-depth Btrfs: Send/Receive

- available with kernel 3.6
- Allows to save the difference between subvolumes
- Use-case 1: Daily backup
 - `btrfs subvolume snapshot -r /orig /orig/Thu`
 - `btrfs send -p /orig/Wed /orig/Thu > Wed-Thu.btrfs`
 - `btrfs receive /backup < Wed-Thu.btrfs`
 - The file `Wed-Thu.btrfs` contains a stream of create, rename, clone, mkdir etc. commands
- Use-case 2: Speed up comparison of snapshots for Snapper

Btrfs Operations 1/2

- **mkfs.btrfs**
 - Different RAID algorithm for data and metadata
 - Different sized disks
- **btrfs-convert**
 - In-place conversion of Ext3 or Ext4 to Btrfs
 - Reversible
- **Balance**
 - Read all extents
 - Pass data through balancer
- **Scrub**
 - Identify and repair data corruption
 - Read all extents and verify checksum
 - In case of problem restore block from mirror (if avail.)
- **Defrag**
 - Re-allocate files to
 - Mount option autodefrag
 - Batch defrag

Btrfs Operations 2/2

- Create subvolume

- `btrfs subvolume create /home`

- Create RO snapshot

- `btrfs subvol snap -r /home/home.`date` -I``

- Roll-back entire snapshot

- “All-or-nothing”

- `mount -o subvol=`

- Atomic operation

- For / fs boot parameter:

- `rootflags=subvol=@/.snapshots /mysnap`

- Roll-back files

- Copy single files from snapshot to “main” filesystem

- No atomic roll-back

Demo 2

- Make filesystem
- Btrfs utility:
 - Create subvolume
 - Create snapshot
 - Start scrub
- Mount subvolume and snapshot

Use Case: Snapper and ITIL

@Begin of implementation Change:

```
snapper create \  
  --type pre \  
  --description "ChgMgt Work order: Upgrade syslog  
configuration to forward log entries to central log  
server" \  
  --userdata \  
  "WorkOrder=201201253030000012-1,  
State=InProgress, Agent=jdoe@example.com"
```

@End of implementation Change:

```
snapper create \  
  --type post --pre-number 240 \  
  --description "Done: ChgMgt Work order: Upgrade syslog  
configuration to forward log entries to central log  
server" \  
  --userdata "WorkOrder=201201253030000012-1,  
State=Closed, Agent=jdoe@example.com"
```

Use Cases For Btrfs

Basic **Btrfs HOWTOs**

- How to create RAID
- Snapshots and subvols
- Grow / shrink

A few others

- System's management:
 - System snapshot and roll-back
 - Pre-patching
- Virtualization
 - Cloned VMs and **containers**
- Data center processes
 - Auditing
 - Change Mgt

A Few Recommendations

Filesystem size:

- Starting: ~30% filling
- Operation: <90% filling

Subvolumes layout

- Directories containing logs to avoid rolling back logs
- High volume directories on different subvolume
- Typically:
/tmp, /srv,
/var/spool,
/var/log,
/var/run,
/var/tmp, /opt

A Few Recommendations

Btrfs on HDD

- Mount options:
 - autodefrag
 - noatime (whenever possible)
- Without “autodefrag” manually defrag on a regular basis!

Btrfs on SSD

- Mount options:
 - discard
 - ssd
 - noatime (whenever possible)
- Disk scheduler: noop
- Never defragment! → wears out SSD

A Few Recommendations 3/3

Filesystem layout

- Depending on system purpose
- Non-mission-critical system:
 - /boot Ext3
 - / Btrfs
 - /db Ext3, ASM, raw
 - /home XFS, Ext3, Btrfs
 - /tmp tmpfs
 - /var Ext3
 - /vmstore XFS, Ext3, Btrfs

Performance

- Simple test:
sustained read/write
 - SSD and HDD
 - Write test
`dd if=/dev/zero of=btrfs-demo-seq-write1 bs=1M count=4096 conv=fsync`
 - Read test
`dd if=btrfs-demo-seq-write1 of=/dev/null bs=1M count=4096 iflag=nocache`

- Results SSD:
 - Seq Write raw: 220 MB/s
 - Seq Write Btrfs: 200 MB/s
 - Seq Read raw: 225 MB/s
 - Seq Read Btrfs: 220 MB/s
- Results HDD:
 - Seq Write Btrfs: 32 MB/s
- For more benchmarking info see:
 - Chris Mason's Btrfs Intro
 - Avi Miller's LinuxConf AU talk
 - Douglas Fuller's [talk](#)

Demo 3

- Convert existing Ext3 to Btrfs
- On-line resize Btrfs
 - Grow
 - Shrink

References

Publications

- Btrfs [wiki](#) (and [mirror](#))
- Josef Bacik's [article](#) on Btrfs
- Arne Jansen's [paper](#) on qgroups (quota support)
- Oloh Rodeh - B-trees, Shadowing, and Clones, IBM Research [paper](#)
- LWN - “A short history of btrfs” [article](#)
- Wikipedia - [Btrfs article](#)

Video's

- Chris Mason: Introduction to Btrfs (26min, [link](#))
- Chris Mason: Btrfs Filesystem: Status and New Features, (May 2012, [link](#))
- Avi Miller's Btrfs [talk](#) at LinuxConf AU (49min, Jan 2012)
 - Demo of “mount -o recovery”
 - Animations of disk usage when creating large nr. of files on Ext3, XFS and Btrfs
- Douglas Fuller's [talk](#) (24min, Apr 2011)
 - Nice performance demo's

Agenda

Introduction to Btrfs

Btrfs in SUSE distro's Snapper

Btrfs in-depth
Use cases

Summary and
Questions

Summary

- Lots of desirable features
- Development is ongoing
- Distributions support is mounting
- **Lots** of practical applications yet to come

For more information please
visit our website:

www.suse.com

Thank you.





Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

